sensors

MDPI

Article

A Robust Diffusion Minimum Kernel Risk-Sensitive Loss Algorithm over Multitask Sensor Networks

Xinyu Li [1,2,†], Qing Shi [2], Shuangyi Xiao [2], Shukai Duan [1,2] and Feng Chen [1,2,*,†]

1 College of Artificial Intelligence, Southwest University, Chongqing 400715, China; lxyv5@email.swu.edu.cn
2 Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, and College of Electronic and Information Engineering, Southwest University, and Chongqing Collaborative Innovation Center for Brain Science, Chongqing 400715, China; shiqing@email.swu.edu.cn (Q.S.); xsy2016@email.swu.edu.cn (S.X.); duansk@swu.edu.cn (S.D.)
* Correspondence: fengchenmit@swu.edu.cn
† Current address: Chongqing Collaborative Innovation Center for Brain Science, Southwest University, Chongqing 400715, China.

Received: 28 April 2019; Accepted: 17 May 2019; Published: 21 May 2019

Abstract: Distributed estimation over sensor networks has attracted much attention due to its various applications. The mean-square error (MSE) criterion is one of the most popular cost functions used in distributed estimation, which achieves its optimality only under Gaussian noise. However, impulsive noise also widely exists in real-world sensor networks. Thus, the distributed estimation algorithm based on the minimum kernel risk-sensitive loss (MKRSL) criterion is proposed in this paper to deal with non-Gaussian noise, particularly for impulsive noise. Furthermore, multiple tasks estimation problems in sensor networks are considered. Differing from a conventional single-task, the unknown parameters (tasks) can be different for different nodes in the multitask problem. Another important issue we focus on is the impact of the task similarity among nodes on multitask estimation performance. Besides, the performance of mean and mean square are analyzed theoretically. Simulation results verify a superior performance of the proposed algorithm compared with other related algorithms.

Keywords: distributed estimation; diffusion minimum kernel risk-sensitive loss; multitask; impulsive noise; sensor networks

1. Introduction

Distributed data processing over sensor networks has emerged as an attractive and challenging research area for various applications such as industrial automation, cognitive radios and inference tasks [1–4]. Distributed estimation plays a significant role in distributed data processing, which estimates some parameters of interest from noise measurements by exchanging information with neighboring nodes. Most algorithms proposed for distributed estimation can be classified into a consensus strategy [5–8], incremental strategy [9–11] and diffusion strategy [12–14]. In our work, we center on a diffusion strategy, which is robust, fully distributed and flexible among these strategies [15–19].

Diffusion strategies are particularly attractive schemes in distributed estimation, such as diffusion Recursive Least Squares (RLS) [20,21], diffusion Least Mean Square (LMS) [13,14]. With the mean-square error (MSE) criterion, these algorithms can accomplish a satisfying performance in a Gaussian noise environment. However, their performance may deteriorate dramatically in the presence of impulsive noise [22,23]. Some algorithms have been proposed to solve the issue, including Diffusion least-mean power (D-LMP) and the Diffusion sign-error Least Mean Square

(DSE-LMS) adaptive filtering algorithm [24,25]. To efficiently address the non-Gaussian noise, the correntropy [26,27] was proposed, which is a higher order statistic and widely used in adaptive filters. Moreover, the generalized maximum correntropy criterion (GMCC) algorithm and the minimum kernel risk-sensitive loss (MKRSL) were proposed [28,29], which provide more general frameworks and better performance. In this work, we consider the diffusion minimum kernel risk-sensitive loss (D-MKRSL) algorithm for distributed estimation over multitask networks.

In previous works, diffusion strategies mainly focus on the single-task estimation problem where an identical parameter vector is estimated by all the nodes [30]. On the contrary, many essential applications are multitask-oriented, such as regression, web page categorization and target location tracking. In the above situations, multiple optimum vectors are different but related, which are inferred synchronously over the networks by all nodes in a collaborative manner. This type of problem is known as a multitask problem. Generally, distributed estimation problems over multitask networks can be roughly classified into two fields. In the first case, there is no knowledge about the correlation of tasks. Meanwhile, which nodes share the same tasks is unknown and nodes cooperate according to network topology [31–33]. In the second situation, it is assumed that nodes know which cluster they belong to and the parameter vector in each cluster is the same. Exploiting the information about the similarity of tasks, diffusion strategies for distributed estimation over multitask are obtained [34–37]. In our work, we focus on the second case.

Inspired by the adapt-then-combine (ATC) DLMS algorithm, we propose the diffusion MKRSL algorithm over multitask networks. The algorithm can achieve desirable performance in both Gaussian and impulsive noise environments. Additionally, the impact of task relatedness on estimation performance is also studied. Moreover, the mean and mean square stability are analyzed theoretically. Effectiveness and advantages of the proposed algorithm are verified by simulation results.

The remaining parts of the article are organized as follows: In Section 2, we briefly introduce the data model of distributed estimation and propose the multitask Diffusion MKRSL algorithm. In Section 3, the mean and mean square performance of the multitask D-MKRSL algorithm are analyzed. Simulation results are demonstrated in Section 4. Finally, we draw conclusions in Section 5.

Notation: We use $(.)^T$, $E[.]$ and $\otimes$ to denote transposition, expectation and Kronecker product operators, respectively. $\mathbf{I}_m$ denotes an $m \times m$ identity matrix. $\mathbf{1}$ is an $N \times 1$ all-unity vector. $|.|$ is the absolute value of a scalar.

2. Multitask Diffusion Estimation

2.1. Data Model

Let us consider a connected network with K nodes. Every node $k \in \{1, 2, ..., K\}$ has access to scalar random variables $d_{k,i}$ and a zero-mean $M \times 1$ regression vector $\mathbf{u}_{k,i}$ at every time instant $i \geq 0$. The data of node k is related via the linear regression model:

$$d_{k,i} = \mathbf{u}_{k,i}^T \mathbf{w}_k^0 + n_{k,i} \tag{1}$$

where $n_{k,i}$ is the random measurement noise with zero-mean and variance $\sigma_{n,k}^2$, which is independent of regression vector $\mathbf{u}_{k,i}$. The goal of distributed estimation is to estimate an $M \times 1$ deterministic but unknown vector $\mathbf{w}_k^0$ by exchanging and combining the data only from neighboring nodes, which is regarded as single-task problem with $w_k^0 = w^0$ for $k = 1, 2, ..., K$, and multitask problem with $w_k^0 \neq w_l^0$ for $k \neq l$. It is assumed that there is no limit to how much information can be transmitted among neighbors.

2.2. Diffusion MKRSL Algorithm

In many previous works, the diffusion distributed estimation algorithms are based on the MSE criterion, which achieves desirable performance if the measurement noise is Gaussian, while their

performance may deteriorate dramatically in an impulsive noise environment. To solve the parameter estimation problem over multitask sensors networks, it becomes a significant focus of our interest to design a novel algorithm that is robust to both Gaussian noises and impulsive noises.

The information theoretic learning (ITL) plays a significant role and provides a general framework in distributed parameter estimation for non-Gaussian cases. The correntropy is a local statistical similarity measure in ITL, which is defined by Reference [26]

$$V(X,Y) = \mathrm{E}[k_\sigma(X-Y)] = \int k_\sigma(x-y)dF_{XY}(x,y) \tag{2}$$

where X, Y are two random variables, $k_\sigma(.)$ is a shift-invariant Mercer kernel and $\sigma > 0$ denotes the kernel bandwidth. $F_{XY}(x,y)$ is the joint distribution function of (X,Y). In our work, we focus on the Gaussian kernel, which takes the following form:

$$k_\sigma(x-y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right) \tag{3}$$

The minimum kernel risk-sensitive loss (MKRSL) algorithm is derived by applying the KRSL to develop a new adaptive filtering algorithm, which shows better convex properties than correntropic loss on the error performance surface [29,38]. The KRSL between two random variables X and Y is defined by

$$\begin{aligned} L_\lambda(X,Y) &= \frac{1}{\lambda}\mathrm{E}[\exp(\lambda(1-k_\sigma(X-Y)))] \\ &= \frac{1}{\lambda}\int \exp(\lambda(1-k_\sigma(X-Y)))dF_{XY}(x,y) \end{aligned} \tag{4}$$

where $\lambda > 0$ is the risk-sensitive parameter. Nevertheless, the exact joint distribution of (X,Y) is usually unavailable in application scenarios. On the contrary, only a limited number of sample values $\{x(i), y(i)\}_{i=1}^{L}$ are known. Therefore, the sample mean estimator of KRSL—called empirical KRSL—is calculated by an average over samples:

$$\hat{L}_\lambda(X,Y) = \frac{1}{L\lambda}\sum_{i=1}^{L}\exp(\lambda(1-k_\sigma(x(i)-y(i)))) \tag{5}$$

Then, the KRSL cost function is derived as

$$J_{KRSL} = \frac{1}{L\lambda}\sum_{i=1}^{L}\exp(\lambda(1-k_\sigma(e(i)))) \tag{6}$$

with $e(i) = d(i) - \mathbf{u}_i^T\mathbf{w}$. The time average of the KRSL cost function in the above equation can be replaced by the expectation

$$J'_{KRSL} = \frac{1}{\lambda}\mathrm{E}[\exp(\lambda(1-k_\sigma(e(i))))] \tag{7}$$

Based on the KRSL cost function mention in the above Equation(7), the instantaneous cost function of the KRSL algorithm is approximated as

$$\tilde{J}_{KRSL} = \frac{1}{\lambda}\exp(\lambda(1-k_\sigma(e(i)))) \tag{8}$$

For the distributed diffusion estimation problem, our goal is to seek the best $\mathbf{w}_k^0$ by minimizing the diffusion KRSL cost function at each node k by cooperating with all neighboring nodes. For each node

k, N_k is the one-hop neighbor set and $\{c_{l,k}\}$ are non-negative real cooperative according to Metropolis rule weights satisfying

$$c_{l,k} = \begin{cases} \dfrac{1}{\max(n_k, n_l)}, \text{ if } l \in N_k \backslash k, \\ 1 - \sum\limits_{l \in N_k \backslash k} c_{l,k}, \text{ if } l = k, \\ 0. \qquad\qquad \text{if } l \notin N_k, \end{cases} \tag{9}$$

where n_k is the degree of node k. The real, non-negative combining coefficients $c_{l,k}$ satisfy the following conditions: $\sum\limits_{l \in N_k \cup k} c_{l,k} = 1$ and $c_{l,k} = 0 \ \ if \ \ l \notin N_k$, $\mathbf{C}\mathbb{I} = \mathbb{I}$, $\mathbb{I}^T\mathbf{C} = \mathbb{I}^T$, where $\mathbf{C}$ is an $N \times N$ matrix. The KRSL local cost function at each node k can be formulated as

$$\begin{aligned} J_k^{loc}(\mathbf{w}) &= \sum_{l \in N_k} c_{l,k} \tilde{J}_{KRSL}(e_{l,i}) \\ &= \frac{1}{\lambda} \sum_{l \in N_k} c_{l,k} \exp(\lambda(1 - k_\sigma(e_{l,i}))) \\ &= \frac{1}{\lambda} \sum_{l \in N_k} c_{l,k} \exp(\lambda(1 - k_\sigma(d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}))) \end{aligned} \tag{10}$$

Based on the KRSL local cost function, the derivative of (10) with respect to w can be derived as

$$\begin{aligned} \nabla J_k^{loc}(\mathbf{w}) &= \frac{1}{\lambda} \sum_{l \in N_k} c_{l,k} \frac{\partial}{\partial w} \exp(\lambda(1 - k_\sigma(e_{l,i}))) \\ &= -\frac{1}{\sigma^2} \sum_{l \in N_k} c_{l,k} \exp(\lambda(1 - k_\sigma(e_{l,i}))) k_\sigma(e_{l,i}) e_{l,i} \mathbf{u}_{l,i}^T \end{aligned} \tag{11}$$

At node k, the weight vector update equation based on a stochastic gradient for $\mathbf{w}_k^0$ is obtained by

$$\begin{aligned} \mathbf{w}_k(i) &= \mathbf{w}_k(i-1) - \mu \nabla J_k^{loc}(\mathbf{w}) \\ &= \mathbf{w}_k(i-1) + \frac{\mu}{\sigma^2} \sum_{l \in N_k} c_{l,k} \exp(\lambda(1 - k_\sigma(e_{l,i}))) k_\sigma(e_{l,i}) e_{l,i} \mathbf{u}_{l,i}^T \\ &= \mathbf{w}_k(i-1) + \eta \sum_{l \in N_k} c_{l,k} \exp(\lambda(1 - k_\sigma(e_{l,i}))) k_\sigma(e_{l,i}) e_{l,i} \mathbf{u}_{l,i}^T \end{aligned} \tag{12}$$

where $\eta = \frac{\mu}{\sigma^2}$ is step-size and $\mathbf{w}_k(i)$ is estimator for $\mathbf{w}_k^0$ at time index i. The above algorithm is a new expression of the MKRSL algorithm. Inspired by the general framework for a diffusion-based distributed estimation algorithm [13], an adapt-then-combine (ATC) strategy for a diffusion MKRSL algorithm is proposed. The ATC scheme first updates the value of the estimator for each node with the adaptive algorithm. Then, the intermediate estimates are fused from its neighbors for each node k. The intermediate estimate at each node k is defined as:

$$\varphi_k(i-1) = \sum_{l \in N_k} \beta_{l,k} \mathbf{w}_l(i-1) \tag{13}$$

The nodes update their intermediate estimates by

$$\varphi_k(i) = \varphi_k(i-1) + \eta \sum_{l \in N_k} c_{l,k} \exp(\lambda(1 - k_\sigma(e_{l,i}))) k_\sigma(e_{l,i}) e_{l,i} \mathbf{u}_{l,i}^T \tag{14}$$

$\varphi_k(i-1)$ is an intermediate estimate at time index $i-1$ for node k. The non-negative real value $\beta_{l,k}$ is a weight coefficient, which corresponds to the matrices $\mathbf{B}$, especially $\mathbf{B} = I$ in the ATC scheme [12]. Therefore, we can obtain:

$$\varphi_k(i) = \mathbf{w}_k(i-1) + \eta \sum_{l \in N_k} c_{l,k} \exp(\lambda(1 - k_\sigma(e_{l,i})))k_\sigma(e_{l,i})e_{l,i}\mathbf{u}_{l,i}^T \tag{15}$$

In the above Equation (15), the task relatedness among nodes is ignored, which is called non-cooperative diffusion MKRSL in this article.

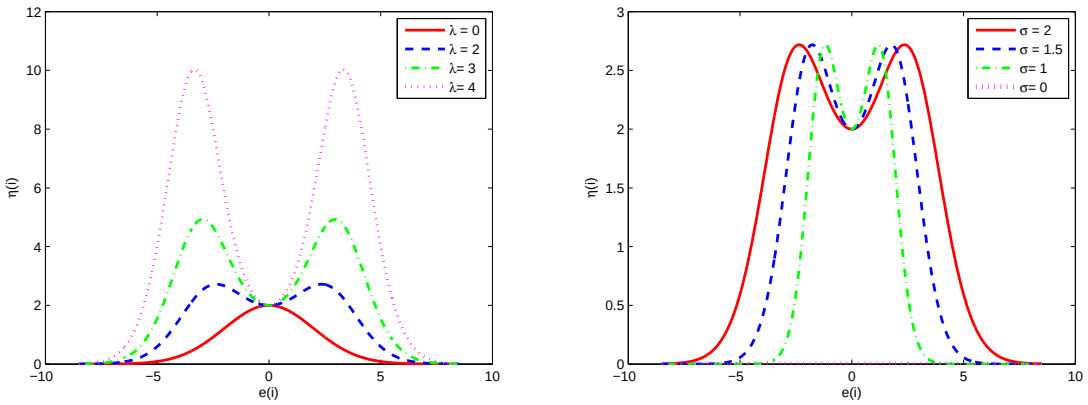
However, multitask estimation is an attracting filed in practical applications. Nodes are grouped into some clusters and each cluster has an identical task in clustered multi-task networks. Furthermore, utilizing the relatedness of tasks, the performance of distributed estimation can be improved. The Equation (15) is adjusted for multitask estimation:

$$\varphi_k(i) = \mathbf{w}_k(i-1) + \eta \sum_{l \in N_k \cap c(k)} c_{l,k} \exp(\lambda(1 - k_\sigma(e_{l,i})))k_\sigma(e_{l,i})e_{l,i}\mathbf{u}_{l,i}^T + \tau \sum_{l \in N_k \setminus c(k)} \rho_{kl}(\mathbf{w}_l(i) - \mathbf{w}_k(i)) \tag{16}$$

$c(k)$ is the cluster of node k, with the cluster of node k non-negative strength parameter τ, weights ρ_{kl} and $\eta(i) = \exp(\lambda(1 - k_\sigma(e_i)))k_\sigma(e_i)$. The notation $N_k \cap c(k)$ is the set of neighboring nodes k and in the same cluster as k. On the contrary, $N_k \setminus c(k)$ denotes the set of neighboring nodes of k that are not in the same cluster as k. The Equations (15) and (16) are defined as the increment step. The combination step can then be derived as

$$\mathbf{w}_k(i) = \sum_{l \in N_k} c_{l,k}\varphi_l(i) \tag{17}$$

The step-size $\eta(i)$ is a function of $e(i)$ and the curves with different values of λ (where $\sigma = \eta = 2.0$) and σ (where $\lambda = \eta = 2.0$) is depicted in Figure 1.



(**a**) (**b**)

Figure 1. Curves of $\eta(i)$ as a function of $e(i)$ (**a**)different values of $\lambda(\sigma = \eta = 2.0)$ (**b**)different values of $\sigma(\lambda = \eta = 2.0)$.

It is shown that the step-size $\eta(i)$ will approach zero as $|e(i)| \to \infty$ for different values of λ. Therefore, the MKRSL algorithm maintains the robustness to outliers, such as impulsive noise.

For a better understanding, the Multitask Diffusion MKRSL algorithm is summarized in Algorithm 1:

Algorithm 1: Multitask Diffusion MKRSL Algorithm

Input: $\mathbf{d}_{k,i}$, $\mathbf{u}_{k,i}^T$, η, τ, and $\{c_{l,k}\}$ satisfying (10)
Initialization: Start with $\{\mathbf{w}_{l,-1} = 0\}$ for all l.
for $i = 1 : T$
for each node k**:**
Adaptation

$$\varphi_k(i) = \mathbf{w}_k(i-1) + \eta \sum_{l \in N_k \cap c(k)} c_{l,k} \exp(\lambda(1 - k_\sigma(e_{l,i})))k_\sigma(e_{l,i}) e_{l,i} \mathbf{u}_{l,i}^T + \tau \sum_{l \in N_k \setminus c(k)} \rho_{kl}(\mathbf{w}_l(i) - \mathbf{w}_k(i))$$

Communication
Transmit the intermediate $\varphi_k(i)$ **to all neighbors in** N_k
Combination

$$\mathbf{w}_k(i) = \sum_{l \in N_k} c_{l,k} \varphi_l(i)$$

end for

3. Performance Analysis

The multitask D-MKRSL algorithms are evaluated theoretically under model (1) in this section. In the following, some common assumptions are adopted for tractable analysis [39,40].

(1) The regression vector $\mathbf{u}_{k,i}$ is independently and identically distributed (i.i.d.) and $E\left[\mathbf{u}_{k,i}\mathbf{u}_{k,i}^T\right] = \mathbf{R}_{u,k}$.

(2) For each node k at time index i, the input noise $n_k(i)$ is independent of $\mathbf{u}_{k,i}$ and is a mixture signal of zero mean Gaussian, we have $\mathrm{E}[n_{k,i}] = 0$.

(3) The step-size η is small enough, so the squared value can be negligible.

Then, the estimate-error vectors are defined as follows:

$$\tilde{\mathbf{w}}_{k,i} = \mathbf{w}_k^0 - \mathbf{w}_{k,i} \tag{18}$$

and

$$\tilde{\varphi}_{k,i} = \mathbf{w}_k^0 - \varphi_{k,i} \tag{19}$$

Furthermore, the global quantities are defined to covert the local variables to global ones:

$$\mathbf{K} = blockdiag\{\eta I_M, ..., \eta I_M\} \tag{20}$$

$$\mathbf{X} = blockdiag\{\tau I_M, ..., \tau I_M\} \tag{21}$$

$$\tilde{\mathbf{w}}_i = col\{\tilde{\mathbf{w}}_{1,i}, ..., \tilde{\mathbf{w}}_{K,i}\} \tag{22}$$

$$\mathbf{w}_i = col\{\mathbf{w}_{1,i}, ..., \mathbf{w}_{K,i}\} \tag{23}$$

$$\mathbf{w}_k^0 = col\left\{\mathbf{w}_1^0, ..., \mathbf{w}_K^0\right\} \tag{24}$$

3.1. Mean Performance

We consider the gradient error caused by replacing the cost function of KRSL with instantaneous values. The gradient error of the intermediate estimate at time i and each node k is defined as follows:

$$\mathbf{s}_k(\mathbf{w}_{k,i-1}) = \hat{\mathbf{f}}_k(\mathbf{w}_{k,i-1}) - \mathbf{f}_k(\mathbf{w}_{k,i-1}) \tag{25}$$

where $\hat{\mathbf{f}}_k(\mathbf{w}_{k,i-1}) = \frac{1}{\sigma^2}\exp(\lambda(1-k_\sigma(e_{k,i-1})))k_\sigma(e_{k,i-1})e_{k,i-1}\mathbf{u}_{k,i-1}^T$ and $\mathbf{f}_k(\mathbf{w}_{k,i-1}) = \frac{1}{\sigma^2}\mathrm{E}[\exp(\lambda(1-k_\sigma(e_{k,i-1})))k_\sigma(e_{k,i-1})e_{k,i-1}\mathbf{u}_{k,i-1}^T]$

The update equation of the intermediate estimate can be rewritten as

$$\varphi_{k,i} = \mathbf{w}_{k,i-1} + \mu\left(\mathbf{s}_k(\mathbf{w}_{k,i-1}) + \mathbf{f}_k(\mathbf{w}_{k,i-1})\right) \tag{26}$$

$\mathbf{f}_k(\mathbf{w}_{k,i-1})$ is twice continuous differentiable in a neighborhood of a line segment between points w_k^0 and w_{k-1}. Thus, based on the Theorem 1.2.1 in Reference [41], we have

$$\mathbf{f}_k(\mathbf{w}_{k,i-1}) = \mathbf{f}_k(\mathbf{w}_k^0) - \left(\int_0^1 H_k\left(\mathbf{w}_k^0 - t\tilde{\mathbf{w}}_{k,i-1}\right)dt\right)\tilde{\mathbf{w}}_{k,i-1} \tag{27}$$

where $H_k(\mathbf{w})$ is the Hessian matrix of $\mathbf{f}_k(\mathbf{w}_{k,i-1})$. $\tilde{\mathbf{w}}_{k,i-1} = \mathbf{w}_k^0 - \mathbf{w}_{k,i-1}$ is the weight error vector for node k. The unknown vector $\mathbf{w}_k^0$ is the real-value that we want to estimate, so $\mathbf{f}_k(\mathbf{w}_k^0)$ is equal to zero. The estimate of each node converges to the vicinity of the unknown vector $\mathbf{w}_k^0$. Thus, $\tilde{\mathbf{w}}_{k,i}$ is small enough such that it is negligible, yielding

$$\begin{aligned}\mathbf{f}_k(\mathbf{w}_{k,i-1}) &\approx -\left(\int_0^1 H_k\left(\mathbf{w}_k^0\right)dt\right)\tilde{\mathbf{w}}_{k,i-1}\\ &= -H_k(\mathbf{w}_k^0)\tilde{\mathbf{w}}_{k,i-1}\\ &= -\beta\mathbf{R}_{u,k}\tilde{\mathbf{w}}_{k,i-1}\end{aligned} \tag{28}$$

where $\mathbf{R}_{u,k}$=$E\left[\mathbf{u}_{k,i}\mathbf{u}_{k,i}^T\right]$ and β is a constant.

So, the approximate value of the gradient error at the value of $\mathbf{w}_k^0$ is

$$\begin{aligned}\mathbf{s}_k(\mathbf{w}_{k,i-1}) &\approx \mathbf{s}_k(\mathbf{w}_k^0)\\ &= \hat{\mathbf{f}}_k(\mathbf{w}_k^0) - \mathbf{f}_k(\mathbf{w}_k^0)\\ &= \frac{1}{\sigma^2}\exp(\lambda(1-k_\sigma(e_{k,i-1})))k_\sigma(e_{k,i-1})e_{k,i-1}\mathbf{u}_{k,i-1}^T\end{aligned} \tag{29}$$

Substituting (28) and (29) into (26) and adjusting for multitask estimation, we can get the intermediate estimate

$$\varphi_{k,i} = \mathbf{w}_{k,i-1} + \mu(\mathbf{s}_k(\mathbf{w}_k^0) - H_k\left(\mathbf{w}_k^0\right)\tilde{\mathbf{w}}_{k,i-1}) + \tau\mathbf{Q}\left(\tilde{\mathbf{w}}_{k,i-1} + \mathbf{w}_k^0\right) \tag{30}$$

where

$$\mathbf{Q}=\mathbf{I_{MN}} - \mathbf{P}\otimes\mathbf{I_M} \tag{31}$$

$\mathbf{P}$ is the matrix with (k,l)-th entry ρ_{kl}. Substituting (30) into (17), we can get the update equation of $\mathbf{w}_k(i)$ as follows

$$\mathbf{w}_k(i) = \sum_{l\in N_k} c_{l,k}[\mathbf{w}_{k,i-1}+\mu(\mathbf{s}_k(\mathbf{w}_k^0) - H_k\left(\mathbf{w}_k^0\right)\tilde{\mathbf{w}}_{k,i-1})+\tau\mathbf{Q}\left(\tilde{\mathbf{w}}_{k,i-1} + \mathbf{w}_k^0\right)] \tag{32}$$

Define global quantity $\mathbf{H} = diag\left\{H_1\left(\mathbf{w}_1^0\right),...,H_k\left(\mathbf{w}_N^0\right)\right\}$ and rewrite (32) as

$$\mathbf{w}_i = \mathbf{C}\left(\mathbf{w}_{i-1} + \mathbf{K}\mathbf{s}_i - \mathbf{K}\mathbf{H}\tilde{\mathbf{w}}_{i-1}+\mathbf{X}\mathbf{Q}\tilde{\mathbf{w}}_{i-1} + \mathbf{X}\mathbf{Q}\mathbf{w}^0\right) \tag{33}$$

Noting that $\mathbf{C}\mathbf{w}^0 = \mathbf{w}^0$, subtracting both sides of (33) from $\mathbf{w}^0$, the global vector is obtained:

$$\tilde{\mathbf{w}}_{i-1} = \mathbf{C}\left(\mathbf{I_{MN}} - \mathbf{K}\mathbf{H} + \mathbf{X}\mathbf{Q}\right)\tilde{\mathbf{w}}_{i-1} + \mathbf{C}\mathbf{K}\mathbf{s}_i + \mathbf{C}\mathbf{X}\mathbf{Q}\mathbf{w}^0 \tag{34}$$

Calculating the expectation of (34) leads to

$$\mathrm{E}\left[\tilde{\mathbf{w}}_{i-1}\right] = \mathbf{C}\left(\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{XQ}\right)\mathrm{E}\left[\tilde{\mathbf{w}}_{i-1}\right] + \mathbf{CK}\,\mathrm{E}\left[\mathbf{s}_i\right] + \mathbf{CXQw}^0 \tag{35}$$

where $\mathrm{E}\left[\mathbf{s}_i\right]$ =$col\left\{E[\mathbf{s}_1(\mathbf{w}_1^0), ..., s_N(\mathbf{w}_N^0)]\right\} = 0$. Based on Lemma 1 of [13], the matrix $\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{X}$ should be stable to guarantee mean stability. There it holds that

$$\left|\lambda_{\max}\left(\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{XQ}\right)\right| < 1 \tag{36}$$

$\lambda_{\max}$ is the largest eigenvalue of matrix. Thus, a sufficient condition for maintaining the stability of the algorithm is:

$$0 < \eta < \frac{2}{\beta\lambda_{\max}\left(\mathbf{R}_{u,k}\right) + 2\tau} \tag{37}$$

3.2. Mean-Square Performance

In this section, we mainly focus on the mean-square performance of the proposed algorithm. Computing the weight norm of (34) and calculating the expectations, we can obtain

$$\mathrm{E}\left[\left\|\tilde{\mathbf{w}}_i\right\|_{\mathbf{\Sigma}}^2\right] = \mathrm{E}\left[\left\|\tilde{\mathbf{w}}_{i-1}\right\|_{\mathbf{\Gamma}}^2\right] + \mathrm{E}[\mathbf{s}_i^T\mathbf{KC}^T\mathbf{\Sigma CKs}_i] + 2\Big(\mathbf{XQw}^0\Big)^T\mathbf{\Sigma C}\left(\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{XQ}\right)\mathrm{E}[\tilde{\mathbf{w}}_{i-1}] + \mathbf{CXQw}^0 \tag{38}$$

where

$$\mathbf{\Gamma} = \left(\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{XQ}\right)\mathbf{C}^T\mathbf{\Sigma C}\left(\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{XQ}\right) \tag{39}$$

and $\mathbf{\Sigma}$ is an Hermitian non-negative-definite matrix. $\tilde{\mathbf{w}}_i$ is dependent of $\mathbf{\Gamma}$ with Assumptions 1 and 2. Therefore, we have:

$$\mathrm{E}\left[\left\|\tilde{\mathbf{w}}_{i-1}\right\|_{\mathbf{\Gamma}}^2\right] = \mathrm{E}\left[\left\|\tilde{\mathbf{w}}_{i-1}\right\|_{\mathrm{E}[\mathbf{\Gamma}]}^2\right] \tag{40}$$

Let

$$\gamma = vec\left\{\mathrm{E}\left[\mathbf{\Gamma}\right]\right\} \tag{41}$$

and

$$\sigma = vec\left\{\mathbf{\Sigma}\right\} \tag{42}$$

where $vec(.)$ is the transpose of the vectorization of a matrix. The Equation (40) can be rewritten to follow equation with (41), (42):

$$\mathrm{E}\left[\left\|\tilde{\mathbf{w}}_i\right\|_{\sigma}^2\right] = \mathrm{E}\left[\left\|\tilde{\mathbf{w}}_{i-1}\right\|_{\gamma}^2\right] + \mathrm{E}[\mathbf{s}_i^T\mathbf{KC}^T\mathbf{\Sigma CKs}_i] + 2\left(\mathbf{XQw}^0\right)^T\mathbf{\Sigma C}\left(\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{XQ}\right)\mathrm{E}[\tilde{\mathbf{w}}_{i-1}] + \mathbf{CXQw}^0 \tag{43}$$

The vectorization operator denoted by Reference [42] is:

$$vec\left\{ABC\right\} = \Big(C^T \otimes A\Big)\,vec\left\{B\right\} \tag{44}$$

Taking expectation and vectorization operations with (38), (41), (42), we have

$$\gamma = \delta\sigma \tag{45}$$

where

$$\delta = E\left[\left(\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{XQ}\right) \otimes \left(\mathbf{I_{MN}} - \mathbf{KH} + \mathbf{XQ}\right)\right]\mathbf{Z} \tag{46}$$

$$\mathbf{Z} = E\left[\mathbf{C}^T \otimes \mathbf{C}^T\right] \tag{47}$$

Based on the relationship of the matrix trace and the vectorization operator [42], we have

$$tr\left\{\mathbf{A}^T\mathbf{B}\right\} = vec^T\left\{\mathbf{B}\right\}vec\left\{\mathbf{A}\right\} \tag{48}$$

$\mathbf{\Sigma}$ is symmetric and deterministic, and we obtain

$$\mathrm{E}[\mathbf{s}_i^T\mathbf{K}\mathbf{C}^T\mathbf{\Sigma}\mathbf{C}\mathbf{K}\mathbf{s_i}]=vec^T\{\mathbf{V}\}\,\mathbf{Z}\sigma \tag{49}$$

where $\mathbf{V}=\mathbf{K}E\left[\mathbf{s}_i\mathbf{s}_i^T\right]\mathbf{K}$. According to A.1 and A.2, $\mathbf{V}$ can be evaluated as

$$\mathbf{V} = blockdiag\left\{\eta^2\mathbf{s}_1^2\left(\mathbf{w}_1^0\right),...,\eta^2\mathbf{s}_K^2\left(\mathbf{w}_K^0\right)\right\} \tag{50}$$

Substitution of (45) and (50) into (43) has

$$\mathrm{E}\left[\|\tilde{\mathbf{w}}_i\|_\sigma^2\right] = E\left[\|\tilde{\mathbf{w}}_{i-1}\|_{\delta\sigma}^2\right] + vec\{\mathbf{V}\}\,\mathbf{Z}\sigma+2\left(\mathbf{X}\mathbf{Q}\mathbf{w}^0\right)^T\mathbf{\Sigma}\mathbf{C}\left(\mathbf{I_{MN}}-\mathbf{K}\mathbf{H}+\mathbf{X}\mathbf{Q}\right)\mathrm{E}[\tilde{\mathbf{w}}_{i-1}]+\mathbf{C}\mathbf{X}\mathbf{Q}\mathbf{w}^0 \tag{51}$$

the recursion of Equation (51) is stable and convergent if the matrix δ is stable. δ can be approximated as

$$\delta \approx \left[\left(\mathbf{I_{MN}}-\mathbf{K}\mathbf{H}+\mathbf{X}\mathbf{Q}\right)\otimes\left(\mathbf{I_{MN}}-\mathbf{K}\mathbf{H}+\mathbf{X}\mathbf{Q}\right)\right]\mathbf{Z} \tag{52}$$

We know that all the entries of $\mathbf{Z}$ are non-negative and all its columns sum up to unity. From the above equation, the stability of δ is in accordance with the stability of $\mathbf{I_{MN}}-\mathbf{K}\mathbf{H}+\mathbf{X}\mathbf{Q}$. Therefore, choosing the step-size lined with the Equation (37) can keep the proposed algorithm stable in the mean-square sense.

4. Simulation

In this section, we validate the performance of the proposed algorithm over multitask sensor networks in two scenarios: a Gaussian environment and an impulsive noise environment. The noise is assumed to be generated by a Gaussian mixture distribution, which is commonly used in signal processing [43,44]:

$$p_{n_i} = (1-v_i)N(0,\sigma_1^2)+v_iN(0,\sigma_2^2) \tag{53}$$

where $N(0,\sigma_i^2)(i=1,2)$ is the Gaussian distribution with zero-mean and variance σ_i^2. And σ_2^2 is set to much larger than σ_1^2, which can generate the impulsive noise.

More frequent impulses are achieved with an increase of v_i, especially

$$\begin{cases} \text{if } v_i = 0 \rightarrow Gaussian \\ \text{if } v_i \neq 0 \rightarrow Impulsive. \end{cases} \tag{54}$$

Increasing v_i leads to more frequent impulses.

We consider a fully connected sensor network with 15 nodes. The network topology and cluster structures are demonstrated in Figure 2. From the network topology, we can easily find that nodes 1 to 6 belong to the first cluster. Meanwhile, nodes 7 to 10 compose the second cluster and nodes 11 to 15 are in the third cluster.

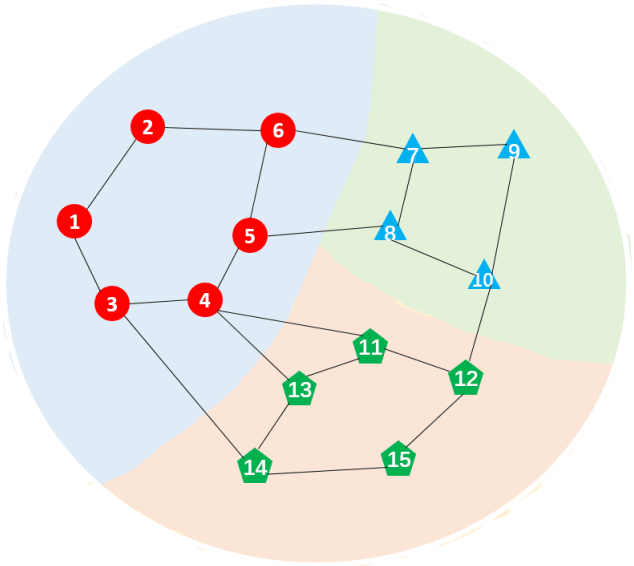


Figure 2. Network topology.

Input variances and noise variances based on Assumptions 1 and 2 are depicted in Figure 3.

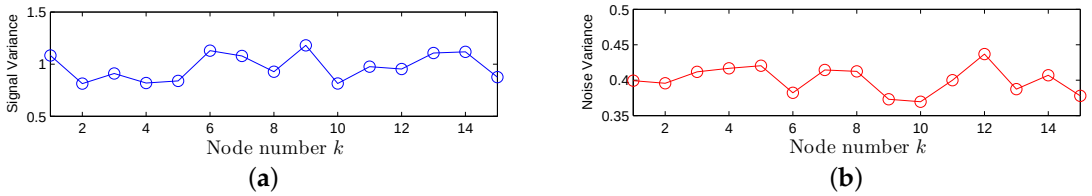
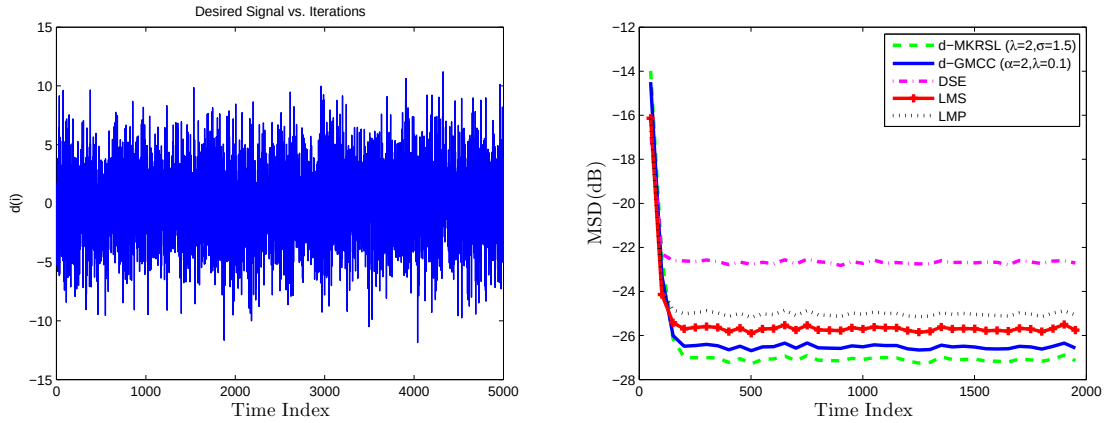


Figure 3. The variances of the input signal (**a**) and noise (**b**).

Scenario 1 (Gaussian noises Environment): As shown in Figure 3, the desired signal is a random process with a zero-mean Gaussian (i.i.d.) noise signal. In the experiment, system parameters are set with $\lambda = 2, \sigma = 1.5$ and the step-size is set with $\eta = 0.02$. τ is a regularization parameter, which promotes similarities between the tasks of the neighboring cluster and is chosen $\tau = 0.1$. The learning curve of the mean square deviation(MSD) is defined as

$$MSD = \frac{1}{K}\sum_{k=1}^{K}\left\|w_k^0 - w_{k,i}\right\|_2^2 \tag{55}$$

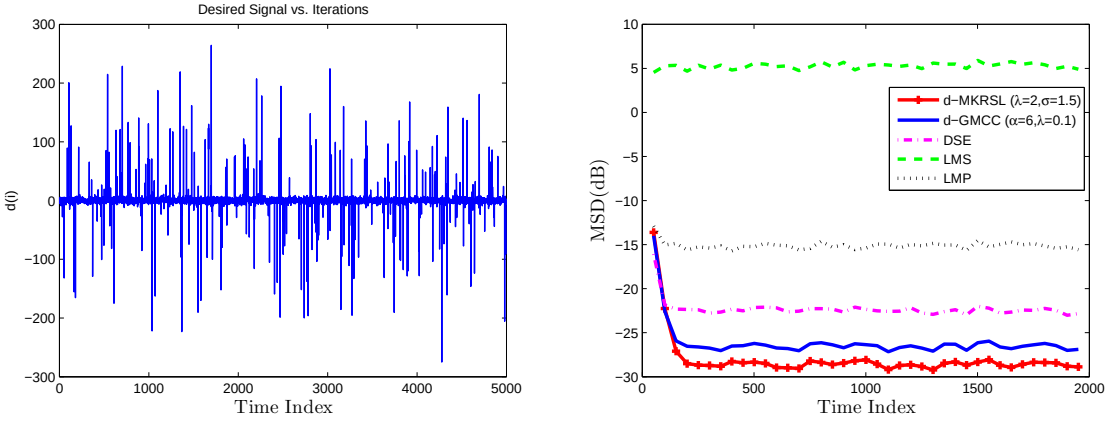
which is adopted for performance comparison. $d(i)$ is the average value of $d_{k,i}$ for all nodes k at time i in Figure 4a. We compare some related algorithms over multitask network, such as diffusion least mean p-power (D-LMP) [21], diffusion generalized maximum correntropy criterion algorithm (D-GMCC) [16], diffusion sign-error LMS (DSE-LMS) [22], D-LMS [12] and the proposed d-MKRSL algorithm in Figure 4b. The step-sizes of all algorithms are chosen after many experiments to ensure the same convergence speed, and other parameters for each algorithm are experimentally selected to achieve a desirable performance. From the above figure, we can conclude that the D-MKRSL algorithm outperforms other related algorithms in the Gaussian noise environment.



(**a**) (**b**)

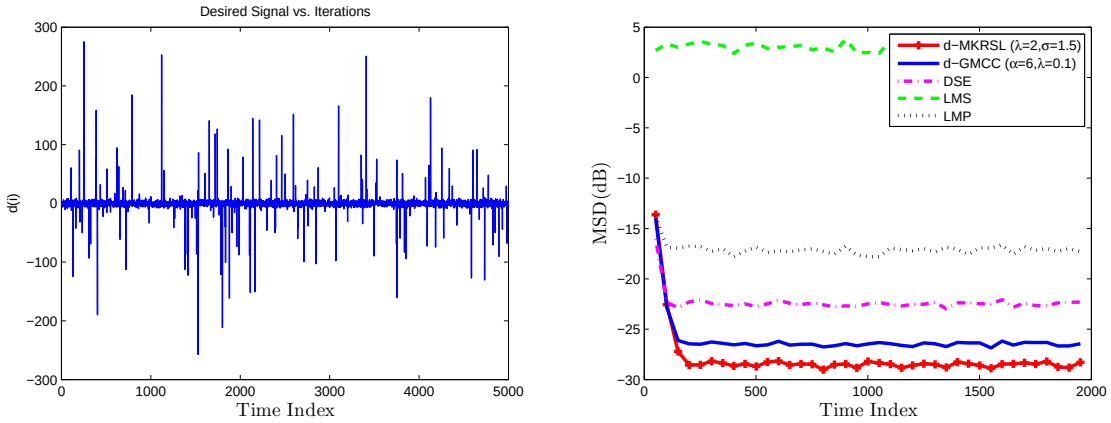
Figure 4. Gaussian noise environment (**a**) desired signal (**b**) transient network MSD(dB).

Scenario 2 (Impulsive noise Environment): The impulsive noise model (54) is adopted to depict the distribution of impulsive interference in the experiment. We now test the influence of the impulsive interference on the performance of the algorithms mentioned above. In Figures 5a and 6a, the desired signals are plotted with $v_i = 0.05, 0.03$ impulsive noise. The corresponding performance of the algorithms in the impulsive noise environment is plotted in Figures 5b and 6b. The value of the parameters α and λ for D-GMCC are selected to achieve the best performance in both the Gaussian and impulsive noise environments. We can observe that the proposed D-MKRSL algorithm is robust and also shows superior performance compared with other related algorithms in the impulsive noise environment.



(**a**) (**b**)

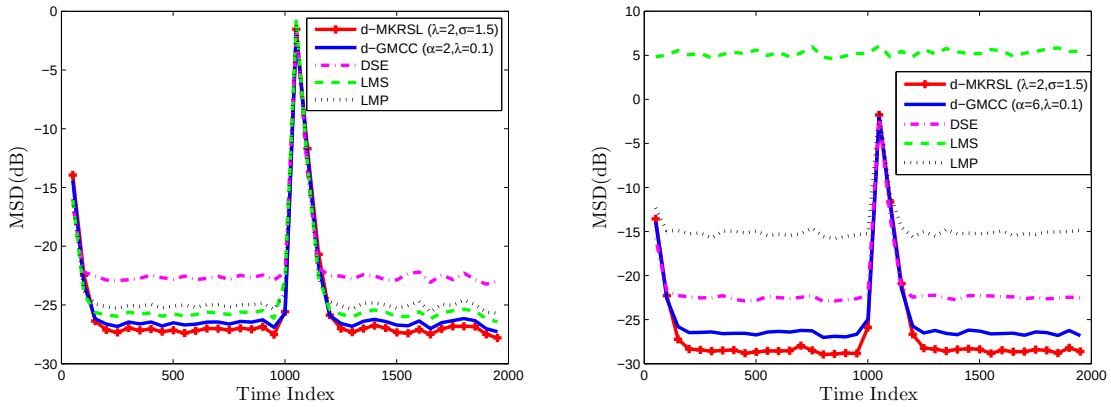
Figure 5. Impulsive interference environment of $v_i = 0.05$ (**a**) desired signal (**b**) transient network MSD(dB).



(**a**) (**b**)

Figure 6. Impulsive interference environment of $v_i = 0.03$ (**a**) desired signal (**b**) transient network MSD(dB).

Furthermore, we consider the performance of the algorithm in a nonstationary scenario and the unknown vector $\mathbf{w}_k^0$ is assumed to change at time 1000. From the convergence curves in Figure 7, it can be easily observed that the proposed algorithm maintains a desirable performance even in the presence of sudden changes of an unknown vector.



(**a**) (**b**)

Figure 7. MSD learning curves in a non-stationary environment (**a**) Gaussian environment (**b**) Impulsive Interference.

Another important aspect is how the correlation of tasks influence the estimation performance. First, we investigate whether the proposed algorithm can promote performance by utilizing the relatedness of tasks compared with non-cooperative strategy. Figure 8 compares the D-MKRSL algorithm with a non-cooperative strategy over a multitask network at identical relatedness of tasks. It is clear that utilizing the relatedness of tasks is beneficial to improve the performance of estimation. Next, the impact of the similarity of tasks on performance is studied. According to Reference [35], the optimum mean vector is assumed to uniformly distribute on a circle of radius r centered at w_k^0. The bigger the value of r is, the smaller the correlation of the tasks will be. Optimum parameter vectors

over the multitask network will be different but related based on the model. The multitask estimation model can be expressed as:

$$w_k^0 = w^0 + r \begin{pmatrix} \cos\theta_k \\ \sin\theta_k \end{pmatrix}$$
$$\theta_k = 2\pi\,(k-1)\,/N + \pi/8 \tag{56}$$

Figure 9 demonstrates that the performance of the algorithms will be improved with the increasing similarity.

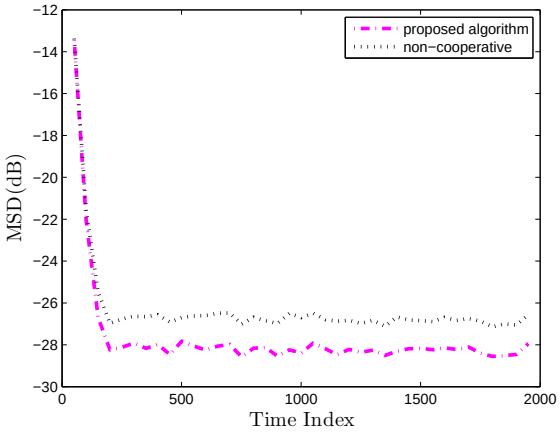


Figure 8. Network MSD comparison over multitask environment.

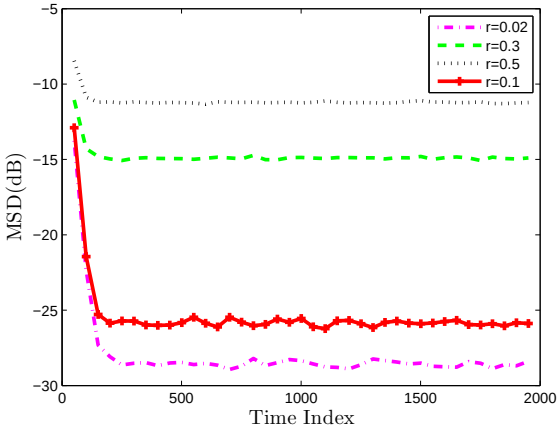


Figure 9. Network MSD comparison with different r value.

5. Conclusions

In this work, we consider the problem of distributed estimation over multitask sensor networks. Then, the D-MKRSL algorithm is proposed and can achieve a desirable performance. Through theoretical analysis, a sufficient condition for ensuring the stability of the D-MKRSL algorithm is obtained. Compared with related algorithms, the simulation results show that the D-MKRSL algorithm has better performance in both Gaussian and impulsive noise environments. Furthermore, we uncover the relationship between the relatedness of tasks and estimation performance. It is demonstrated that the performance is improved with a higher correlation among tasks by cooperation strategy.

Author Contributions: Data curation, F.C.; Funding acquisition, S.D.; Project administration, X.L.; Software, X.L. and Q.S.; Supervision, S.D. and F.C.; Writing–original draft, X.L.; Writing–review and editing, Q.S. and S.X.

Funding: This work was supported in part by the National Natural Science Foundation of China (Grant No.61875168) and Chongqing Research Program of Basic Research and Frontier Technology (No. cstc2017jcyjAX0265).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sayed, A.H. Adaptation, learning, and optimization over networks. *Found. Trends Mach. Learn.* **2014**, *7*, 311–801. [CrossRef]
2. Lorenzo, P.D.; Barbarossa, S.; Sayed, A.H. Bio-inspired swarming for dynamic radio access based on diffusion adaptation. In Proceedings of the 2011 19th European Signal Processing Conference (EUSIPCO), Barcelona, Spain, 29 August–2 September 2011; pp. 402–406.
3. Chen, J.; Cao, X.; Cheng, P.; Xiao, Y.; Sun, Y. Distributed collaborative control for industrial automation with wireless sensor and actuator networks. *IEEE Trans. Ind. Electron.* **2010**, *57*, 4219–4230. [CrossRef]
4. Sayed, A.H.; Tu, S.; Chen, J.; Zhao, X.; Towfic, Z.J. Diffusion strategies for adaptation and learning over networks. *IEEE Signal Process. Mag.* **2013**, *30*, 155–171. [CrossRef]
5. Olfati-Saber, R.; Fax, J.A.; Murray, R.M. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **2007**, *95*, 215–233. [CrossRef]
6. Kar, S.; Moura, J.M.F. Distributed consensus algorithms in sensor networks: Link failures and channel noise. *IEEE Trans. Signal Process.* **2009**, *57*, 355–369. [CrossRef]
7. Wang, J.; Peng, D.; Jing, Z.; Chen, J. Consensus-Based Filter for Distributed Sensor Networks with Colored Measurement Noise. *Sensors* **2018**, *18*, 3678. [CrossRef] [PubMed]
8. Nedic, A.; Ozdaglar, A. Distributed subgradient methods for multiagent optimization. *IEEE Trans. Autom. Control* **2009**, *54*, 48–61. [CrossRef]
9. Nedic, A.; Bertsekas, D.P. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. Optim.* **2001**, *12*, 109–138. [CrossRef]
10. Rabbat, M.G.; Nowak, R.D. Quantized incremental algorithms for distributed optimization. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 798–808. [CrossRef]
11. Lopes, C.G.; Sayed, A.H. Incremental adaptive strategies over distributed networks. *IEEE Trans. Signal Process.* **2007**, *48*, 223–229. [CrossRef]
12. Chen, J.; Sayed, A.H. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process.* **2012**, *60*, 4289–4305. [CrossRef]
13. Cattivelli, F.S.; Sayed, A.H. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.* **2010**, *58*, 1035–1048. [CrossRef]
14. Zhao, X.; Sayed, A.H. Performance limits for distributed estimation over LMS adaptive networks. *IEEE Trans. Signal Process.* **2012**, *60*, 5107–5124. [CrossRef]
15. Tu, S.Y.; Sayed, A.H. Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Trans. Signal Process.* **2012**, *60*, 6217–6234. [CrossRef]
16. Chen, F.; Li, X.; Duan, S.; Wang, L.; Wu, J. Diffusion generalized maximum correntropy criterion algorithm for distributed estimation over multitask network. *Digit. Signal Process.* **2018**, *81*, 16–25. [CrossRef]
17. Liu, Y.; Li, C.; Tang, W.K.S.; Zhang, Z. Distributed estimation over complex networks. *Inf. Sci.* **2012**, *197*, 91–104. [CrossRef]
18. Chen, F.; Shao, X. Broken-motifs diffusion LMS algorithm for reducing communication load. *Signal Process.* **2017**, *197*, 91–104. [CrossRef]
19. Chen, F.; Shao, X. Complementary performance analysis of general complex-valued diffusion LMS for noncircular signals. *Signal Process.* **2019**, *160*, 237–246.
20. Cattivelli, F.S.; Lopes, C.G.; Sayed, A.H. A diffusion RLS scheme for distributed estimation over adaptive networks. In Proceedings of the 2007 IEEE 8th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Helsinki, Finland, 17–20 June 2007; pp. 1–5.
21. Cattivelli, F.S.; Lopes, C.G.; Sayed, A.H. Diffusion recursive leasts-quares for distributed estimation over adaptive networks. *IEEE Trans. Signal Process.* **2008**, *56*, 1865–1877. [CrossRef]

22. Gao, W.; Chen, J. Kernel Least Mean p-Power algorithm. *IEEE Signal Process. Lett.* **2017**, *24*, 996–1000. [CrossRef]
23. Shao, X.; Chen, F.; Ye, Q.; Duan, S. A Robust Diffusion Estimation Algorithm with Self-Adjusting Step-Size in WSNs. *Sensors* **2017**, *17*, 824. [CrossRef] [PubMed]
24. Wen, F. Diffusion least-mean P-power algorithms for distributed estimation in alpha-stable noise environments. *Electron. Lett.* **2013**, *49*, 1355–1356. [CrossRef]
25. Ni, J.; Chen, J.; Chen, X. Diffusion sign-error LMS algorithm: formulation and stochastic behavior analysis. *Signal Process.* **2016**, *128*, 142–149. [CrossRef]
26. Liu,W.; Pokharel, P.P.; Principe, J.C. Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [CrossRef]
27. Chen, B.; Liu, X.; Zhao, H.; Principe, J.C. Maximum correntropy Kalman filter. *Automatica* **2017**, *76*, 70–77. [CrossRef]
28. Chen, B.; Xing, L.; Zhao, H.; Zheng, N.; Principe, J.C. Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **2016**, *64*, 3376–3387. [CrossRef]
29. Chen, B.; Xing, L.; Xu, B.; Zhao, H.; Zheng, N.; Principe, J.C. Kernel Risk-Sensitive Loss: Definition, Properties and Application to Robust Adaptive Filtering. *IEEE Trans. Signal Process.* **2017**, *65*, 2888–2901. [CrossRef]
30. Chen, J.; Richard, C.; Sayed, A.H. Diffusion LMS over multitask networks. *IEEE Trans. Signal Process.* **2015**, *63*, 2733–2748. [CrossRef]
31. Chen, J.; Sayed, A.H. Distributed Pareto optimization via diffusion strategies. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 205–220. [CrossRef]
32. Zhao, X.; Sayed, A.H. Clustering via diffusion adaptation over networks. In Proceedings of the 2012 3rd International Workshop on Cognitive Information Processing (CIP), Parador de Baiona, Spain, 28–30 May 2012; pp. 1–6.
33. Zhao, X.; Sayed, A.H. Distributed clustering and learning over networks. *IEEE Trans. Signal Process.* **2015**, *63*, 3285–3300. [CrossRef]
34. Chen, J.; Richard, C.; Hero, A.O.; Sayed, A.H. Diffusion LMS for mu1titask problems with overlapping hypothesis subspaces. In Proceedings of the 2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Reims, France, 21–24 September 2014; pp. 1–6.
35. Bogdanovic, N.; Plata-Chaves, J.; Berberidis, K. Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 7223–7227.
36. Chen, J.; Richard, C.; Sayed, A.H. Multitask diffusion adaptation over networks. *IEEE Trans. Signal Process.* **2014**, *62*, 4129–4144. [CrossRef]
37. Nassif, R.; Richard, C.; Ferrari, A. Proximal multitask learning over networks with sparsity-inducing coregularization. *IEEE Trans. Signal Process.* **2016**, *64*, 6329–6344. [CrossRef]
38. Ma, W.; Chen, B.; Duan, J.; Zhao, H. Diffusion maximum correntropy criterion algorithms for robust distributed estimation. *Digit. Signal Process.* **2016**, *58*, 10–19. [CrossRef]
39. Sayed, A.H. *Adaptive Filters;* Wiley: Hoboken, NJ, USA, 2008.
40. Haykin, S. *Adaptive Filter Theory;* Prentice-Hall: Upper Saddle River, NJ, USA, 2002.
41. Kelley, C.T. *Iterative Methods for Optimization;* SIAM: Philadelphia, PA, USA, 1999.
42. Abadir, K.M.; Magnus, J.R. *Matrix Algebra*; Cambridge University Press: Cambridge, UK, 2005.
43. Chan, S.C.; Zou, Y.X. A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: fast algorithm and convergence performance analysis. *IEEE Trans. Signal Process.* **2004**, *52*, 975–991. [CrossRef]
44. Sayed, A.S.; Zoubir, A.M.; Sayed, A.H. Robust adaptation in impulsive noise. *IEEE Trans. Signal Process.* **2016**, *64*, 2851–2865. [CrossRef]